

Distributed

**FREE**  
to over 4,000 Pick  
installations!

**PRAGMA'S**

# PRODUCT PROFILES

News and Information for Pick™ Operating System Users

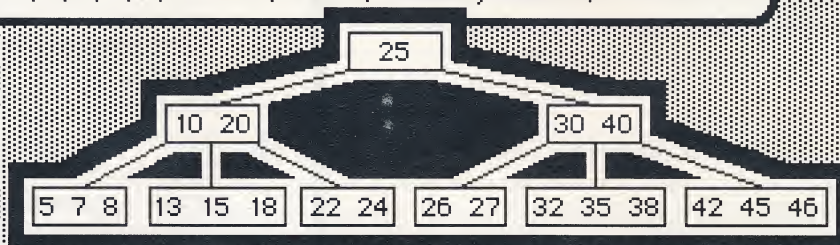
Issue Number 24

March 1986

|           |       |                        |                     |       |
|-----------|-------|------------------------|---------------------|-------|
| D POLL    | 5251  | SCOTTYS HOUSPAINTING   | 2010 ADMIRAL WAY    | 96001 |
| J RIZZO   | 5250  | ZENITH COMPUTER INC    | 377 CORONADO DR     | 96003 |
|           | 12822 | MIRROR POOLS           | 6 SUZANNE LN        | 96003 |
| M SHEID   | 2575  |                        | 14255 OKANOGAN CT   | 96120 |
| J SKULLY  | 6955  | UNIV OF CALIFORNIA     | 19700 ELISA AVE     | 96271 |
| M STEWART | 4408  | SALT & PEPPER MFG      | 20774 MEADOW OAK RD | 96328 |
| K VANN    | 17377 | EARTHTECHNICS          | BOX 2687            | 96701 |
| M ZEISLY  | 9038  | SUNSCOPE RESOURCES     | 629 MAIN ST         | 96706 |
| R HIDE    | 6285  | WESTECH FINANCIAL SVCS | 1757-B JUNCTION AVE | 96720 |
| B STERNS  | 4478  | WESTECH FINANCIAL SVCS | 1757-B JUNCTION AVE | 96720 |
| S LUKAUW  | 5362  |                        | 470 N 3RD ST #2     | 96720 |
|           | 9746  | VIKING PUBLISHING      | 777 N 1ST ST #600   | 96725 |
| M RUSSELL | 4529  | RARE EDITIONS LTD      | 2115 EUREKA WAY     | 96740 |
|           | 10992 | SEAGATE TECHNOLOGIES   | 1175 HILLTOP DR     | 96743 |

N/U/UU/D/DD/W1/W2/W3/Z/C/L/X/return/chars/lname<;fname>/<. >id?

*Beyond the  
Power of  
Pick,  
Revisited*



207 GRANADA DRIVE  
APTOS, CA 95003

BULK RATE  
U.S. POSTAGE  
PAID  
APTOS, CA 95003  
Permit No. 67

**Address Correction Requested**



## Pragma's PRODUCT PROFILES

Issue #24 • March 1986

*Product Profiles* is published periodically by:

**Semaphore Corporation**  
207 Granada Drive  
Aptos, CA 95003

Telephone 408-688-9200

Entire contents copyright © 1986 by Semaphore Corporation • All rights reserved • No part of this publication may be reproduced in any form or by any means without prior written permission from Semaphore Corporation • Semaphore offers no warranty, either express or implied, for any losses due to the use of any material published in *Product Profiles* • Subscriptions are free, but are available only to Pick operating system users in the USA • Back issues are available by sending a stamped, self-addressed envelope • All material received will be considered for publication • Semaphore reserves the right to edit all submittals • Please tell us your mailing label number when calling or writing • Semaphore cannot be responsible for missing or lost issues • Send your new address at least four weeks before moving • Pick is a trademark of Pick Systems • Semaphore Corporation and its publications are not affiliated with Pick Systems.

### Does Your Mailing Label Say RENEW?

If the mailing label on this issue says  
RENEW, your free subscription has expired.  
To avoid missing the next issue, renew  
immediately by returning this coupon!

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

City/State/ZIP: \_\_\_\_\_

Telephone: \_\_\_\_\_

I use the Pick operating system and my mailing address is in the USA. Please continue sending me *Pragma's Product Profiles* free of charge.

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

Return this form to:  
Pragma • 207 Granada Drive • Aptos, CA 95003

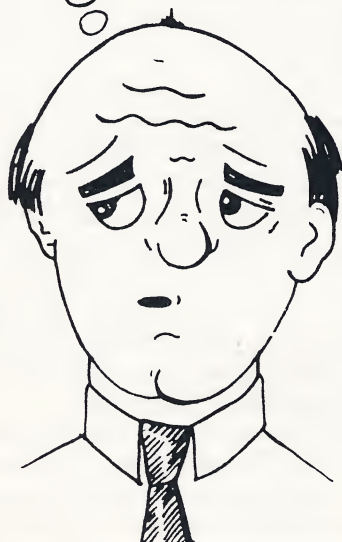
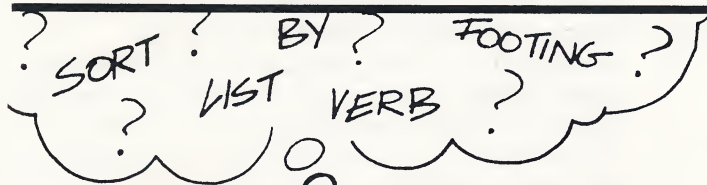
## Beyond the Power of Pick, Revisited

Our January 1985 issue of *Pragma's Product Profiles* featured an article titled *Beyond the Power of the Pick System*, which described some software we had developed at Semaphore Corporation to allow interactive file searches using display terminals in a convenient, user-friendly fashion that made up for some of the inadequacies in the Pick filing system. That original article described how we used doubly-linked lists and inverted cross-reference files to support a "browser" program called FIND, allowing us to search and display our customer files in name, company, or ZIP code order. We could select any sort order at any time, and a display showing columns of names, ID numbers, companies, addresses, and ZIP codes would always be immediately updated on our terminals. Any name or ZIP could instantly be located and displayed upon entry of any 1, 3, or 5 character substring, and the display of customers could be scrolled up or down in various increments at any time. (See this issue's cover for an example of a typical FIND display.)

We also described how the FIND program worked in conjunction with our name and address input program called GET, so that if a customer's name or address changed, all data was immediately deleted from its old position in the sorted display list, and reinserted in the appropriate new position. The result was a Pick system able to instantly sort, search, and browse up and down through a displayed list of any number of file items, all continuously maintained in an arbitrary, operator-selected sort order.

Although our FIND and GET programs greatly enhanced user productivity and were a major improvement over the classical Pick technique of doing a SORT and paging through a long printout or display, we eventually found a few areas we wanted to improve on even more. For one thing, the linked lists and inverted cross-reference files consumed more disk space than we would have liked. Another problem was that the cross-reference file could only be used to locate existing data. For example, we couldn't make the display instantly jump to ZIP codes starting at 950 if such a ZIP wasn't already on file. Another problem was that five characters were often not enough to quickly zero in on a particular item. For example, to find the name SMITHSON when limited to a five character search required starting at SMITH and manually paging through all SMITHs to reach SMITHSON. Another problem was the unpredictable time required to insert a new item. For example, if the cross-reference file happened to already index some existing item with the same ZIP code as a new item, then that was a good place to start searching for the correct insertion point of the new item. But what if the new item's ZIP wasn't already in the cross-reference list? Then the search had to start from some nearby location.





## THE KEY TO ACCESS SUCCESS ON A PICK SYSTEM IS THE ACCESS MANAGER.

You already know what a great tool ACCESS is. It lets you get at your data and make your own reports and inquiries. BUT — wouldn't it be even greater if it were more USER-FRIENDLY? With the ACCESS MANAGER, say good-bye to forgotten words and re-typing the same sentences time after time. Say hello to on-line dictionary display, sentence libraries and instant data verification.

### INSTANT DATA VERIFICATION:

The Access Manager instantly checks the sentence for valid dictionary names and correct sentence construction. You only need re-type from the bad part.

### ON-LINE DICTIONARY:

You can display your file dictionary's name, size and column title.

### FULL ON-LINE HELP:

One command evokes help messages to guide you through correct access sentence construction.

### SENTENCE LIBRARIES:

Store your access sentence for re-execution at a later time. The storage library keeps an index of descriptive phrases for easy retrieval.

### FAULT-TOLERANT SENTENCE LOCATION:

Retrieve your sentence from the library by typing in a descriptive phrase. If the phrase is similar to an existing phrase, the Access Manager will find your sentence.

### COMPLETE DOCUMENTATION:

And of course, the Access Manager comes with complete documentation.

### LABELS:

Construct list-label and sort-label sentences painlessly.

If you have a PICK system with the Execute capability from Basic and need to get at your data, then you can have the ACCESS MANAGER. Please state your Hardware Manufacturer, current software release, and type of tape drive.



**datafaction** 731 S. Highway 101, Suite 2L, Solana Beach, CA 92075 • (619) 755-1155

# YES

PLEASE SEND ME THE ACCESS MANAGER. Enclosed is my check/money order made payable to: DATAFACTION in the amount of \$395.00 (California residents please add appropriate sales tax.)

NAME \_\_\_\_\_

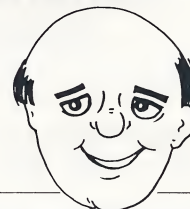
COMPANY \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_

STATE \_\_\_\_\_ ZIP \_\_\_\_\_

PHONE (     ) \_\_\_\_\_



As a result, the response time for a first-time insertion was somewhat unpredictable and inconsistent, which tended to frustrate operators, even if the insertion time never varied by more than a few seconds.

We eventually decided to overhaul and redesign our browser software by using B-trees, a technique we happened to mention at the end of our original article. A B-tree is a data structure for maintaining file items in a certain order. A B-tree consists of *nodes* linked by *pointers*. Usually, each node in a B-tree of order  $n$  contains from  $n$  to  $2n$  keys, and  $n+1$  to  $2n+1$  pointers. Keys are the indices by which we want to retrieve the file's records. Keys might be item IDs, or they might be

any other attribute or fragment of data from the item being indexed. To maintain a B-tree, we simply insert or delete an item's key in the appropriate B-tree node whenever we create or delete an item in the corresponding file being indexed. Unlike the Pick filing system, which can only quickly find a file item if we know the item ID, we can quickly find *any* key in a B-tree. B-trees therefore provide a way to find any file item using any attribute, whether or not we know the item ID. B-trees are searched by comparing the key we're looking for with a key in a node. If our key is less than the node key, the pointer to the left of the key is taken to the next node to continue the search; if our key is greater, the right pointer is taken. (See this issue's cover for an example of a small B-tree of order 2.)

Fortunately, B-trees have the property of staying *balanced*, which means all pointer-less *leaf* nodes at the lowest level in the tree are all at the same depth, with equal-length pointer-paths. As a result, locating or adding a key never involves unpredictable overhead, and there are never any extremely long pointer chains to traverse before finding a key.

B-tree insertions of new keys are accomplished by starting at the uppermost *root* node, comparing keys, and then following the appropriate pointers left or right until a leaf node is found. If there is less than  $2n$  keys already in the node, the new key is simply inserted in the node at the proper position among the other keys. But if the node is already full of keys, it is *split*: the smallest  $n$  keys are placed in one node, the largest  $n$  keys are placed in another node, and the remaining median key is promoted to the parent node. If the parent is also already full, it too is split. In the worst case, splitting propagates all the way back up to the root and the tree increases in height by one level.

B-tree deletions of existing keys are accomplished by first searching from the root in the usual manner to locate the node containing the key to be removed. If the key is not in a leaf node, but in a node with pointers to other sibling nodes, the key is first swapped with the next sequential key, which is the first key in the leftmost node of the subtree pointed to by the found key's right pointer. Once the desired key is (possibly swapped into and) deleted from a leaf node, at least  $n$  keys must remain. If there are less than  $n$  keys left, keys from a neighboring node are moved in to restore the tree's balance and the property that every non-root node has at least  $n$  keys. If the node and its neighbor don't have at least  $2n$  keys to divide between them, then all the keys are merged into one node and the remaining empty node is discarded. Like splitting, merging can propagate back to the root, which shrinks the B-tree by one level.

Once we had programmed the algorithms for B-tree insertion and deletion, it was fairly simple to adapt our name and address software to rely on B-trees instead of linked lists and inverted files. We began by creating an empty file called BTREE for holding B-tree nodes of order 50. No changes to our customer files were necessary. Now when our GET input program creates a

# ZEBRA

HARDWARE & PERIPHERALS

SO CAL FIELD SERVICE  
NATIONAL DEPOT SERVICE

**ATKIN/JONES**  
COMPUTER SERVICE

(714) 953-4351

WE SPECIALIZE IN USED AND COMPATIBLE

**MICRODATA**

SYSTEMS - PERIPHERALS - MEMORY - PRINTERS

50 MB REFLEX I DISC DRIVES  
128 MB REFLEX II DISC DRIVES  
INTELLIGENT 8-WAY BOARDS, CABLES  
STANDARD 8-WAY BOARDS, CABLES  
128K MOS MEMORY BOARDS  
16K CORE MEMORY BOARDS

ADDS VIEWPOINT CRTS  
\*\*\*\*\*  
COMPLETE MOS SYSTEMS  
\*\*\*\*\*

**PRINTRONIX**

PRINTERS

\*\*\*\*\*  
BOARD REPAIR AND EXCHANGE SERVICE AVAILABLE

Call CHUCK HAAS For Competitive Prices  
Buy - Trade - Sell 513-528-5400

**EASTCOMP II INC.**





# The Password Is Prof



## Enter the World of PICK Through a Complete Electronic Learning System

**PROF** is a hands-on, interactive learning system that will get you up to speed on PICK, quickly and easily. It's clever, entertaining, and all learning is done right on the terminal.

**PROF** leads you through simulated, HANDS-ON WORK SESSIONS in which you actively participate. You try every important command, and carefully review the way the system responds to each. It all happens right on the terminal. There is no book to hold on your lap.

**PROF** contains material that is suitable for both users and technical personnel. Give it to your application users. Use it to train new analysts and programmers.

**PROF** allows you to learn at your own pace. You can follow the suggested order of instruction, or quickly jump from "page" to "page"... just like a book.

**PROF** has an Electronic Master Index which works just like the index at the back of a book. You can peruse the alphabetical list of subjects, then jump right into any lesson you desire.

**PROF** tells you exactly what to type. It then "checks" off each character as you type the right key. If you touch the wrong key, you are reminded by a single "beep." Every keystroke is screened electronically. You cannot make a mistake.

**PROF** covers all important aspects of PICK. This includes:

|        |                   |
|--------|-------------------|
| TCL    | Database Concepts |
| EDITOR | PROC              |
| ACCESS | DATA/BASIC        |

**PROF** is customized. There is a different version for each manufacturer of PICK-based equipment, including:

|           |                    |
|-----------|--------------------|
| Ultimate  | General Automation |
| Microdata | Evolution          |
| ADDS      | Pertec             |
| CDI       | Altos              |
| Datamedia | PICK PC-XT         |

YES! Please send me more information about **PROF**.

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Phone (       ) \_\_\_\_\_

You can order **PROF** by writing or calling us at

**CRESCENDO Associates, Inc.**

24350 JOY ROAD, SUITE 7  
REDFORD TOWNSHIP, MICHIGAN 48239  
(313) 537-1919



# FREE Back Issues!

A few back issues of *Pragma's Product Profiles* are still available while supplies last. To receive your FREE copies, indicate which issues you need and send a stamped, self-addressed envelope to:

**Pragma • 207 Granada Dr. • Aptos CA 95003**

(Allow 1.5 oz. per issue to compute postage.)

- Issue #7:** Manufacturing Users And Vendors Wanted
- Issue #8:** A Look At ScreenPlay
- Issue #10:** Beyond The Power Of The Pick System
- Issue #11:** Open Architecture Status Report
- Issue #12:** Our Impressions Of The Zebra 750
- Issue #13:** What Should Standard Pick Be?
- Issue #14:** Our Third Annual Pick Hardware Survey
- Issue #15:** The Wonders Of A Software Upgrade
- Issue #17:** A READ Sometimes Fails
- Issue #19:** New Pick Book A Disappointment
- Issue #20:** uniVerse, A New Home For Pick
- Issue #21:** Programming For Speed
- Issue #22:** How Files Grow
- Issue #23:** GA About To Release 3.2

## PICK SPECIALISTS

SAVINGS UP TO 50% OFF MANUFACTURERS LIST PRICE

**SELL  
BUY  
LEASE  
TRADE**

**(415) 490-8733**

**ADDS-MENTOR  
ULTIMATE  
MICRODATA  
GA-ZEBRA  
PRIME  
PRINTRONIX  
CIE-PRINTERS**

**SYSTEMS—PERIPHERALS—PRINTERS**

- **New and Used Equipment**
- **Immediate Delivery**
- **Upgrades and Spares**

**WE NEED TO BUY YOUR  
USED EQUIPMENT/PLEASE CALL**



**BAYSIDE**  
COMPUTER CORPORATION

**(415) 490-8733**  
43537 Ocaso Corte  
Fremont, CA 94539

new customer, it calls a 130-line INSERT subroutine that automatically places the customer item ID in the proper place in the BTREE file of nodes. (The BTREE file actually contains three root nodes and three whole B-trees. One B-tree keeps customer IDs sorted BY COUNTRY BY ZIP BY ADDRESS BY COMPANY BY LAST-NAME BY FIRST-NAME BY ID, one B-tree keeps customer IDs sorted BY COMPANY BY LAST-NAME BY FIRST-NAME BY ID, and one BY LAST-NAME BY FIRST-NAME BY ID.) When a customer is deleted, GET simply calls the 170-line DELETE subroutine to adjust the B-tree and delete the item ID. Whenever a customer name or address is changed, GET calls DELETE then INSERT to move the customer ID from its old position to the appropriate new position in the B-tree.

The use of B-trees has paid off handsomely. Editing the customer file is on the average faster than with our old cross-reference system, and the automatic B-tree updates made by the INSERT and DELETE subroutines are consistently the same in speed regardless of what portion of the B-tree is being adjusted. One of our name and address files has just over 10,000 items occupying 800,000 bytes, and the equivalent B-tree file requires 157,000 bytes. For another customer file of over 40,000 names and addresses, the corresponding B-tree occupies 723,000 bytes, far less than the linked lists and inverted files we tried before, while offering much more flexibility in searching.

To allow our FIND browser to work with the B-tree file, we also wrote a 60-line LOOKUP subroutine to find the position of any item in the B-tree. An operator can now input a name or address fragment of *any* length, even combinations of different attribute fragments, and FIND will now call LOOKUP to immediately find and display that item. Even if the given fragment doesn't exist in the file, LOOKUP will indicate the B-tree position where such an item *would* be, so now FIND can be used to search and browse on any key, existing or not.

We've been using B-trees in a production environment for many months now, and find them to be fast, efficient, reliable, and easily adaptable to any file. Our users are now spoiled with the ability to instantly search, display, and browse around any value or fragment in any attribute in any file for which a B-tree is maintained, and maintaining a B-tree simply requires a subroutine call or two any time an item is created, changed, or deleted. As a result, our computer system now regularly performs in a fashion significantly beyond the power of a typical Pick machine, and our users never again have to bother waiting for lengthy SORTs and LISTs. Δ

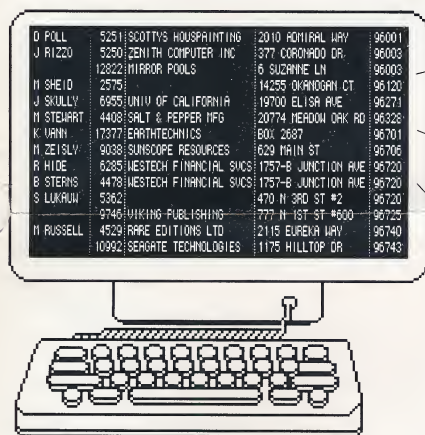
*NOTE: Because of the large number of requests we received asking to purchase our software after we published our previous article about browsing programs, we have packaged our new B-tree software for sale. See the advertisement on the opposite page for details on how to let your computer go beyond the power of a typical Pick system.*



**Tired of waiting  
for your Pick  
computer to  
SORT, SELECT,  
or LIST?**

**Tired of  
redisplaying a  
whole report just  
because you need  
to go back a page?**

*Now you can use **B-TREE-P** to  
instantly view any selected  
items from any Pick file!*



Immediately locate any item in any file just by typing one or more starting characters that match any attribute in the item!

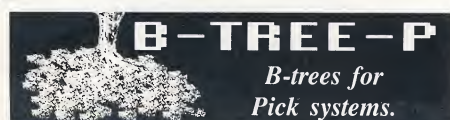
You can display not only a selected item or attribute, but also the previous and next items using any sort order you specify!

You can jump to any item in a file at any time, then browse through the file by scrolling up or down, an item at a time, or a page at a time!

B-TREE-P is a proven collection of subprograms for using B-trees on Pick computer systems. B-trees allow any of the data in any of your Pick files to be instantly located and displayed in any sort order, without having to wait for SORT or SELECT commands.

B-TREE-P and a few minor modifications to your existing data entry programs are all that is necessary for you to immediately be able to search, display, and browse through your data quickly and conveniently.

*No modifications to your existing data files are necessary!*



*B-TREE-P includes all necessary source code for a B-tree system that works with any file:*

- Insertion subroutine
- Deletion subroutine
- Lookup subroutine
- Previous/next subroutine
- Complete instructions

*Plus, you receive the source code for a complete demonstration system that uses B-TREE-P to maintain a name and address file:*

**Editor program** for creating and changing names and addresses.

**Browser program** for displaying names and addresses.

**Printer program** for listing file items in order without having to wait for a sort.

### ***Here's how to order:***

Send your name, address, telephone number, and your check for \$395 payable to Semaphore Corporation to:

*Semaphore Corporation  
207 Granada Drive  
Aptos, CA 95003*

We'll send you complete B-TREE-P source code listings and all necessary documentation.

**WARNING:** B-TREE-P includes a license agreement with copy, use, and transfer restrictions limiting your use of B-TREE-P to one computer.



Made for  
PICK!!

SERIAL COMMUNICATION  
BOARD

STREAMING TAPE

1.2 MB  
FLOPPY  
DISK  
DRIVE

HARD DISK

EXPANSION  
BOARD SLOTS

**ALTOS**  
COMPUTER SYSTEMS  
WORLD LEADER IN  
MULTI-USER MICROSYSTEMS

- Altos 3068 Features!
- Microprocessor —  
32 Bit MC68020
  - Cache Memory —  
on chip: 256 Bytes  
on board: 8 KB 2-  
set associative data  
in instruction cache
  - Bus —  
32 Bit data bus
  - Main Memory —  
1-8 MB
  - Hard Disk —  
Up to 195 MB  
Formatted  
Track caching
  - Streaming Tape —  
60 MB:  
90 IPS
  - Floppy Disk —  
1.2 MB

## The 68020-based Altos 3068.

If you're a PICK software developer or dealer,  
you owe it to yourself to see a demonstration  
of the forty-user Altos 3068.

Call this number, and we'll make  
the arrangements.  
**(800) ALTOS U.S.**

PICK is a trademark of Pick Systems.

**ALTOS**<sup>®</sup>

COMPUTER SYSTEMS  
2641 Orchard Parkway  
San Jose, CA 95134

**WE SUPPORT PICK.**